Extensions to the `ltxdoc` class *†

Arthur Ogawa ‡

June 21, 2022

This file embodies the `ltxdocext` package, the implementation and its user documentation.

The distribution point for this work is `journals.aps.org/revtex`, which contains prebuilt runtime files, documentation, and full source, ready to add to a TDS-compliant TEX installation.

To use this document class, you must have a working TEX installation equipped with LATEX 2_ε and possibly pdftex and Adobe Acrobat Reader or equivalent.

To install, retrieve the distribution, unpack it into a directory on the target computer, and move the files `ltxdocext.sty` and `acrofont.sty` into a location in your filesystem where they will be found by LATEX.

If you will be using the `acrofont` package, you must also install the virtual fonts `zpsynocmrv`, `zptmnocmr`, `zptmnocmrm`, and `zpzcnocmry`. The corresponding `.tfm`, `.vf`, and `.vpl` files are part of this distribution.

To use, read the user documentation `ltxdocext.pdf`. The `.dtx` file, `ltxdocext.dtx`, constitutes in itself an instance of use of the `ltxdocext` package and the `acrofont` package.

Contents

*This file has version number 1.0a, last revised 2018/12/26.

‡`mailto:arthur_ogawa at sbcglobal.net`

1 Processing Instructions

The package files `ltxdocext.sty` and `acrofont.sty` are generated from this file, `ltxdocext.dtx`, using the DOCSTRIP facility of LaTeXvia `tex ltxdocext.dtx` (Note: do *not* use LaTeX for this step). The typeset documentation that you are now reading is generated from the same file by typesetting it with LaTeX or pdftex via `latex ltxdocext.dtx` or `pdflatex ltxdocext.dtx`.

1.1 Build Instructions

You may bootstrap this suite of files solely from `ltxdocext.dtx`. Prepare by installing LaTeX 2$_\varepsilon$ (and either tex or pdftex) on your computer, then carry out the following steps:

1. Within an otherwise empty directory, typeset `ltxdocext.dtx` with TeX or pdftex; thereby generating the package file `ltxdocext.sty`, and the package file `acrofont.sty`. Make sure that DOCSTRIP receives permission to overwrite existing versions of these packages.

2. Now typeset `ltxdocext.dtx` with LaTeX or pdflatex; you will obtain the typeset documentation you are now reading, along with the file `README`.

   Note: you will have to run LaTeX, then `makeindex -s gind.ist ltxdocext.idx`, then `makeindex -s gglo.ist -o ltxdocext.gls ltxdocext.glo`, then LaTeX again in order to obtain a valid index and table of contents.

3. Install the following files into indicated locations within your TDS-compliant `texmf` tree (you may need root access):

   - `$TEXMF/tex/latex/revtex/ltxdocext.sty` and `$TEXMF/tex/latex/revtex/acrofont.sty`
   - `$TEXMF/source/latex/revtex/ltxdocext.dtx`
   - `$TEXMF/doc/latex/revtex/ltxdocext.pdf`

   where `$TEXMF/` stands for `texmf-local/`, or some other `texmf` tree in your installation.

4. Run `mktexlsr` on `$TEXMF/` (you may need root access).

5. Build and installation are now complete; now put a `\usepackage{ltxutil}` in your document preamble!

1.2 Change Log

1.3 Bill of Materials

Following is a list of the files in this distribution arranged according to provenance.

1.3.1 Primary Source

One single file generates all.

```
%ltxdocext.dtx
%
```

1.3.2 Generated by `latex ltxdocext.dtx`

Typesetting the source file under pdflatex generates the readme and the documentation.

```
%README ltxdocext.pdf
%
```

1.3.3 Generated by `tex ltxdocext.ins`

Typesetting this file with TEX generates the package file.

```
%ltxdocext.sty    acrofont.sty
%
```

1.3.4 Documentation

The following are the online documentation:

```
%ltxdocext.pdf
%
```

1.3.5 Auxiliary

The following are auxiliary files generated in the course of running LATEX:

```
%ltxdocext.aux ltxdocext.idx ltxdocext.ind ltxdocext.log ltxdocext.toc
%
```

2 Code common to all modules

We want to require only one place in this file where the version number is stated, and we also want to ensure that the version number is embedded into every generated file.

Now we declare that these files can only be used with LATEX 2_ε. An appropriate message is displayed if a different TEX format is used.

```
%<*driver|package|fonts>
\NeedsTeXFormat{LaTeX2e}[1995/12/01]%
%</driver|package|fonts>
```

As desired, the following modules all take common version information:

```
%<package>\ProvidesFile{ltxdocext.sty}%
%<fonts>\ProvidesFile{acrofont.sty}%
%<*driver>
\expandafter\ProvidesFile\expandafter{\jobname.dtx}%
%</driver>
```

The following line contains, for once and for all, the version and date information. By various means, this information is reproduced consistently in all generated files and in the typeset documentation.

```
%<version>
 [2018/12/26 1.0a ltxdoc extensions package]% \fileversion
```

3 The driver module `doc`

This module, consisting of the present section, typesets the programmer's documentation, generating the `README-LTXDOCEXT` as required.

Because the only uncommented-out lines of code at the beginning of this file constitute the `doc` module itself, we can simply typeset the `.dtx` file directly, and there is thus rarely any need to generate the "doc" DOCSTRIP module. Module

delimiters are nonetheless required so that this code does not find its way into the other modules.

The `\end{document}` command concludes the typesetting run.

```
%<*driver>
```

3.1 The Preamble

The programmers documentation is formatted with the `ltxdoc` document class, with local customizations, and with the usual code line indexing.

```
\documentclass[draft]{ltxdoc}
\RequirePackage{ltxdocext}%
\RequirePackage[colorlinks=true,linkcolor=blue]{hyperref}%
%\expandafter\ifx\csname package@font\endcsname\@undefined\else
% \expandafter\RequirePackage\expandafter{\csname package@font\endcsname}%
%\fi
\CodelineIndex\EnableCrossrefs % makeindex -s gind.ist ltxdocext
\RecordChanges % makeindex -s gglo.ist -o ltxdocext.gls ltxdocext.glo
```

3.1.1 Docstrip and info directives

We use so many DOCSTRIP modules that we set the `StandardModuleDepth` counter to 1.

```
\setcounter{StandardModuleDepth}{1}
```

The following command retrieves the date and version information from this file.

```
\expandafter\GetFileInfo\expandafter{\jobname.dtx}%
```

3.2 The "Read Me" File

As promised above, here is the contents of the "Read Me" file. That file serves a double purpose, since it also constitutes the beginining of the programmer's documentation. What better thing, after all, to have appear at the beginning of the typeset documentation?

A good discussion of how to write a ReadMe file can be found in Engst, Tonya, "Writing a ReadMe File? Read This" *MacTech* October 1998, p. 58.

Note the appearance of the `\StopEventually` command, which marks the dividing line between the user documentation and the programmer documentation.

The usual user will not be asked to do a full build, not to speak of the bootstrap. Instructions for carrying these processes begin the programmer's manual.

```
\begin{filecontents*}{README-LTXDOCEXT}
\title{%
 Extensions to the \classname{ltxdoc} class%
 \thanks{%
  This file has version number \fileversion,
  last revised \filedate.%
 }%
 \thanks{%
  Version \fileversion\ \copyright\ 2019 American Physical Society
 }%
}%
\author{%
```

```
Arthur Ogawa%
 \thanks{\texttt{mailto:arthur\_ogawa at sbcglobal.net}}%
}%
%\iffalse
% For version number and date,
% search on "\fileversion" in the .dtx file,
% or see the end of the README file.
%\fi

\maketitle

This file embodies the \classname{ltxdocext} package,
the implementation and its user documentation.

The distribution point for this work is
\url{journals.aps.org/revtex},
which contains prebuilt runtime files, documentation, and full source,
ready to add to a TDS-compliant \TeX\ installation.

The \classname{ltxdocext} package was commissioned by the American Physical Society
and is distributed under the terms of the \LaTeX\ Project Public License,
the same license under which all the portions of \LaTeX\ itself are distributed.
Please see \url{http://ctan.tug.org/macros/latex/base/lppl.txt} for details.

To use this document class, you must have a working
\TeX\ installation equipped with \LaTeXe\
and possibly pdftex and Adobe Acrobat Reader or equivalent.

To install, retrieve the distribution,
unpack it into a directory on the target computer,
and move the files \file{ltxdocext.sty} and \file{acrofont.sty}
into a location in your filesystem where they will be found by \LaTeX.

If you will be using the \classname{acrofont} package, you must
also install the virtual fonts
\file{zpsynocmrv}, \file{zptmnocmr},
\file{zptmnocmrm}, and \file{zpzcnocmry}.
The corresponding \file{.tfm}, \file{.vf}, and \file{.vpl}
files are part of this distribution.

To use, read the user documentation \file{ltxdocext.pdf}.
The \file{.dtx} file, \file{ltxdocext.dtx}, constitutes
in itself an instance of use of the \classname{ltxdocext}
package and the \classname{acrofont} package.

\tableofcontents

\section{Processing Instructions}

The package files \file{ltxdocext.sty} and \file{acrofont.sty}
are generated from this file, \file{ltxdocext.dtx},
using the {\sc docstrip} facility of \LaTeX
via |tex ltxdocext.dtx| (Note: do \emph{not} use \LaTeX\ for this step).
The typeset documentation that you are now reading is generated from
```

```
the same file by typesetting it with \LaTeX\ or pdftex
via |latex ltxdocext.dtx| or |pdflatex ltxdocext.dtx|.

\subsection{Build Instructions}

You may bootstrap this suite of files solely from \file{ltxdocext.dtx}.
Prepare by installing \LaTeXe\ (and either tex or pdftex) on your computer,
then carry out the following steps:
\begin{enumerate}
\item
Within an otherwise empty directory,
typeset \file{ltxdocext.dtx} with \TeX\ or pdftex;
thereby generating the package file \file{ltxdocext.sty},
and the package file \file{acrofont.sty}.
Make sure that {\sc docstrip} receives permission
to overwrite existing versions of these packages.
\item
Now
typeset \file{ltxdocext.dtx} with \LaTeX\ or pdflatex;
you will obtain the typeset documentation you are now reading,
along with
the file \file{README}.

Note: you will have to run \LaTeX, then
\file{makeindex} \texttt{-s gind.ist ltxdocext.idx}, then
\file{makeindex} \texttt{-s gglo.ist -o ltxdocext.gls ltxdocext.glo}, then
\LaTeX\ again in order to obtain a valid index and table of contents.
\item
Install the following files into indicated locations within your
TDS-compliant \texttt{texmf} tree (you may need root access):
\begin{itemize}
\item
\file{$TEXMF/}\file{tex/}\file{latex/}\file{revtex/}\classname{ltxdocext.sty} and
\file{$TEXMF/}\file{tex/}\file{latex/}\file{revtex/}\classname{acrofont.sty}
\item
\file{$TEXMF/}\file{source/}\file{latex/}\file{revtex/}\classname{ltxdocext.dtx}
\item
\file{$TEXMF/}\file{doc/}\file{latex/}\file{revtex/}\classname{ltxdocext.pdf}
\end{itemize}
where \file{$TEXMF/} stands for \file{texmf-local/}, or some other \texttt{texmf} tree
in your installation.
\item
Run \texttt{mktexlsr} on \file{$TEXMF/} (you may need root access).
\item
Build and installation are now complete;
now put a \cmd\usepackage\texttt{\{ltxutil\}} in your document preamble!
\end{enumerate}

\subsection{Change Log}
\changes{1.0a}{2018/12/12}{(MD) Updated name of README file and use standard fonts when types


\end{filecontents*}
```

3.3 The Document Body

Here is the document body, containing only a `\DocInput` directive—referring to this very file. This very cute self-reference is a common `ltxdoc` idiom.

```
\begin{document}%
\expandafter\DocInput\expandafter{\jobname.dtx}%
\PrintChanges
\end{document}
%</driver>
```

4 Using the `ltxdoc` and `acrofont` packages

These packages are an adjunct to the standard LaTeX `ltxdoc` class and may be simply invoked as follows:

```
%\documentclass[draft]{ltxdoc}
%\RequirePackage{ltxdocext}%
%\RequirePackage{acrofont}%
%\CodelineIndex\EnableCrossrefs
%
```

Your document should simply cleave to the standards of the `ltxdoc` class, with extensions and alterations as noted.

4.1 Extensions to the `ltxdoc` class

4.1.1 Extensions to the `verbatim` environment and `\verb` command

The delimiters `<<` and `>>` within the scope of the verbatim environment or within the argument of the `\verb` command produce italics surrounded by angle brackets. This typographic convention usually indicates *metalanguage*, i.e., a placeholder.

To obtain the angle bracket character per se, double the character, viz., "`the delimiter \verb+<<<<+`".

4.1.2 The `-matter` Commands Work

The sectioning commands `\frontmatter`, `\mainmatter`, and `\backmatter`of the standard LaTeX `book` class are operative in the `ltxdoc` class.

4.1.3 The `\GetFileInfo` command

You can use the `\GetFileInfo` command to extract the date, version, and file info of a file which has registered itself via the `\ProvidesFile` or `\ProvidesClass` command (employing the optional argument thereto).

For instance, if your document contains the following:

```
%\RequirePackage{ltxdocext}%
%\GetFileInfo{ltxdocext.sty}%
%
```

then the following control sequence names will have a value corresponding to that package's `\ProvidesFile` command: `\filedate`: the file's date, `\fileversion`: the file's version, and `\fileinfo`: the file's info.

4.1.4 Self-Indexing Commands

Certain commands automatically produce an index entry (or several related entries) according to the meaning.

meta-text	`\marg{`⟨*text*⟩`}`
command	`\cmd\csname`
environment name	`\env{`⟨*name*⟩`}`
`\begin{foo}`	`\envb{`⟨*foo*⟩`}`
`\end{foo}`	`\enve{`⟨*foo*⟩`}`
argument	`\arg{`⟨*name*⟩`}`
optional	`\oarg{`⟨*name*⟩`}`
filename	`\file{`⟨*name*⟩`}`
url	`\url{`⟨*name*⟩`}`
document class	`\classname{`⟨*name*⟩`}`
document substyle	`\substyle{`⟨*name*⟩`}`
class option	`\classoption{`⟨*name*⟩`}`

4.1.5 Unnumbered Tables

When your documentation requires the use of an unnumbered table, use the `unnumtable` environment:

```
%\begin{unnumtable}
%\begin{tabular}{ll}
%⟨table rows⟩
%\end{tabular}
%\end{unnumtable}
%
```

4.1.6 Structuring Tables

The commands `\toprule`, `\colrule`, and `\botrule`allow you to mark the beginning of the column heads the beginning of the table body, and the end of the table body, respectively. In context,

```
%\begin{tabular}{ll}
%\toprule
%⟨table head rows⟩
%\colrule
%⟨table rows⟩
%\botrule
%\end{tabular}
%
```

4.1.7 A Sectioning Command Below `\subsection`

The `\subsubsection` command is defined.

4.2 Alterations to the `ltxdoc` class

The following involve no new markup, but they do change the appearance of your formatted documentation:

1. Using the `acrofont` package causes your document to be formatted using the standard Acrobat fonts to the greatest extent possible. This means that for most documents, Computer Modern is not used at all. Math that unavoidable must use CM still exists, however.
2. An index will be produced at the end of the document without your needing to explicitly mark it up, and it will have an entry in the TOC.
3. The `quote` environment has a slightly smaller left margin.
4. Array columns are set tight by default.
5. A host of `\DoNotIndex` directives are invoked. I intend this list to grow to encompass even more commands. Send me your suggestions.

5 Extensions to the ltxdoc class

The `package` DOCSTRIP module comprises the package `ltxdocext.sty`, which provides extensions to the standard LaTeX `ltxdoc` class.

5.1 Beginning of the `package` DOCSTRIP module

This portion of code is only present in the LaTeX package (`.sty` file), not in the kernel portion.

```
%<*package>
\def\class@name{ltxdocext}%
\expandafter\PackageInfo\expandafter{\class@name}{%
 An extension to the \protect\LaTeXe\space ltxdoc class
 by A. Ogawa (arthur\_ogawa sbcglobal.net)%
}%
%</package>
```

5.2 Beginning of the `kernel` DOCSTRIP module

The bulk of the code is the kernel portion; a brief tail of `package` code then follows.

```
%<*kernel>
```

5.3 Incorporate `ltxguide.cls` extensions

Code extracted from `ltxguide.cls`, by Alan Jeffrey. "This code stolen from `ltxguide.cls`: Some hacks with verbatim... NB: this would be better done with the verbatim package, but this document has to run on any LaTeXinstallation."

```
\RequirePackage{verbatim}%
\let\o@verbatim\verbatim
\def\verbatim{%
  \ifhmode\unskip\par\fi
% \nopagebreak              % Overridden by list penalty
  \ifx\@currsize\normalsize
     \small
  \fi
  \o@verbatim
}%
```

Here we extend the font-setting command to include making <> active (i.e., adjusting the input encoding).

```
\renewcommand \verbatim@font {%
  \normalfont \ttfamily
  \catcode`\<=\active
  \catcode`\>=\active
}%
```

Make |...| a synonym for \verb|...|.

```
\RequirePackage{shortvrb}
\AtBeginDocument{%
 \MakeShortVerb{\|}%
}%
```

Make active bracket characters produce italics surrounded by angle brackets (used in verbatim and \verb). << produces a less-than, and >> produces a greater-than.

```
\begingroup
  \catcode`\<=\active
  \catcode`\>=\active
  \gdef<{\@ifnextchar<\@lt\@meta}
  \gdef>{\@ifnextchar>\@gt\@gtr@err}
  \gdef\@meta#1>{\marg{#1}}
  \gdef\@lt<{\char`\<}
  \gdef\@gt>{\char`\>}
\endgroup
\def\@gtr@err{%
   \ClassError{ltxguide}{%
      Isolated \protect>%
   }{%
      In this document class, \protect<...\protect>
      is used to indicate a parameter.\MessageBreak
      I've just found a \protect> on its own.
      Perhaps you meant to type \protect>\protect>?
   }%
}
\def\verbatim@nolig@list{\do\`\do\,\do\'\do\-}
```

End of code stolen from ltxguide.cls. Thanks, Alan.

Add functionality from doc.dtx: (code stolen from doc.dtx):

```
\def\GetFileInfo#1{%
  \def\filename{#1}%
  \def\@tempb##1 ##2 ##3\relax##4\relax{%
    \def\filedate{##1}%
    \def\fileversion{##2}%
    \def\fileinfo{##3}}%
  \edef\@tempa{\csname ver@#1\endcsname}%
  \expandafter\@tempb\@tempa\relax? ? \relax\relax}
```

(end of code stolen from doc.dtx. Thanks FMi.)

Various forms of self-indexing commands:

```
\DeclareRobustCommand{\marg}[1]{%
 \meta{#1}%
 \index{#1=\string\meta{#1} placeholder}\index{placeholder>#1=\string\meta{#1}}%
}%
```

```
\DeclareRobustCommand\meta[1]{%
 \mbox{\LANGLE\itshape#1\/\RANGLE}%
}%
\def\LANGLE{$\langle$}%
\def\RANGLE{$\rangle$}%
\DeclareRobustCommand{\arg}[1]{%
 {\ttfamily\string{}\meta{#1}{\ttfamily\string}}%
 \index{#1=\string\ttt{#1}, argument}\index{argument>#1=\string\ttt{#1}}%
}%
\let\oarg\undefined
\DeclareRobustCommand{\oarg}[1]{%
 {\ttfamily[%]
  }\meta{#1}{\ttfamily%[
 ]}%
 \index{#1=\string\ttt{#1}, optional argument}%
 \index{argument, optional>#1=\string\ttt{#1}}%
}%
\DeclareRobustCommand\cmd{\begingroup\makeatletter\@cmd}%
\long\def\@cmd#1{%
 \endgroup
 \cs{\expandafter\cmd@to@cs\string#1}%
 \expandafter\cmd@to@index\string#1\@nil
}%
\def\cmd@to@cs#1#2{\char\number`#2\relax}%
\def\cmd@to@index#1#2\@nil{%
 \index{#2=\string\cmd#1#2}%\index{command>#2=\string\cmd#1#2}%
}%
\DeclareRobustCommand\cs[1]{{\ttfamily\char`\\#1}}%
\def\scmd#1{%
 \cs{\expandafter\cmd@to@cs\string#1}%
 \expandafter\scmd@to@index\string#1\@nil
}%
\def\scmd@to@index#1#2\@nil#3{%
 \index{\string$#3=\string\cmd#1#2---#3}%
%\index{command>\string$#3=\string\cmd#1#2---#3}%
}%
\DeclareRobustCommand\env{\name@idx{environment}}%
\DeclareRobustCommand\envb[1]{%
 {\ttfamily\string\begin\string{}\env{#1}{\ttfamily\string}}%
}%
\DeclareRobustCommand\enve[1]{{\ttfamily\string\end\string{}\env{#1}{\ttfamily\string}}}%
\DeclareRobustCommand{\file}{\begingroup\@sanitize\@file}%
\long\def\@file#1{\endgroup
 {\ttfamily#1}%
 \index{#1=\string\ttt{#1}}\index{file>#1=\string\ttt{#1}}%
}%
\DeclareRobustCommand\substyle{\name@idx{document substyle}}%
\DeclareRobustCommand\classoption{\name@idx{document class option}}%
\DeclareRobustCommand\classname{\name@idx{document class}}%
\def\name@idx#1#2{%
 {\ttfamily#2}%
 \index{#2\space#1=\string\ttt{#2}\space#1}\index{#1>#2=\string\ttt{#2}}%
}%
\DeclareRobustCommand\url@ltxdocext{\begingroup\catcode`\/\active\catcode`\.\active\catcode`\
```

```
\AtBeginDocument{%
 \ifx\url\undefined\let\url\url@ltxdocext\fi
}%
\def\@url#1{%
 \url@break{\ttfamily#1}%
 \url@char\edef\@tempa{#1=\string\url{#1}}%
 \expandafter\index\expandafter{\@tempa}%
 \expandafter\index\expandafter{\expandafter u\expandafter r\expandafter l\expandafter >\@tem
 \endgroup
}%
{\catcode`\:\active\aftergroup\def\aftergroup:}{\active@colon}%
\def\colon@break{\colon@char\allowbreak}%
\def\colon@char{:}%
{\catcode`\/\active\aftergroup\def\aftergroup/}{\active@slash}%
\def\slash@break{\slash@char\allowbreak}%
\def\slash@char{/}%
{\catcode`\.\active\aftergroup\def\aftergroup.}{\active@dot}%
\def\dot@break{\dot@char\allowbreak}%
\def\dot@char{.}%
\def\url@break{\let\active@slash\slash@break\let\active@dot\dot@break\let\active@colon\colon@
\def\url@char{\let\active@slash\slash@char\let\active@dot\dot@char\let\active@colon\colon@cha
```

5.4 Changes to the base class of the ltxdoc class

Modify `theindex` environment so that it produces a TOC entry

```
\renewenvironment{theindex}
                {\if@twocolumn
                   \@restonecolfalse
                 \else
                   \@restonecoltrue
                 \fi
                 \columnseprule \z@
                 \columnsep 35\p@
\def\see##1##2{\textit{See} ##1}%
\def\seealso##1##2{\textit{See also} ##1}%
\long\def\cmd##1{\cs{\expandafter\cmd@to@cs\string##1}}%
\def\@url##1{\url@break\ttt{##1}\endgroup}%
\def\ttt{\begingroup\@sanitize\ttfamily\@ttt}%
\def\@ttt##1{##1\endgroup}%
\mathchardef\save@secnumdepth\c@secnumdepth
\c@secnumdepth\m@ne
                 \twocolumn[\section{\indexname}]%
%                \@mkboth{\MakeUppercase\indexname}%
%                        {\MakeUppercase\indexname}%
\c@secnumdepth\save@secnumdepth
                 \thispagestyle{plain}\parindent\z@
                 \parskip\z@ \@plus .3\p@\relax
                 \let\item\@idxitem}
                {\if@restonecol\onecolumn\else\clearpage\fi}
\renewenvironment{quote}
                {\list{}{%
                 \leftmargin1em\relax
                 \rightmargin\leftmargin
                 }%
```

```
                 \item\relax}
                {\endlist}
```

5.5 Extensions to the base class of `ltxdoc.cls`

Matter commands from `book.cls`

```
\newif\if@mainmatter
\newif\if@openright
\@openrighttrue
\DeclareRobustCommand\frontmatter{%
  \cleartorecto
  \@mainmatterfalse
  \pagenumbering{roman}%
}%
\DeclareRobustCommand\mainmatter{%
  \cleartorecto
  \@mainmattertrue
  \pagenumbering{arabic}%
}%
\DeclareRobustCommand\backmatter{%
  \if@openright
    \cleartorecto
  \else
    \clearpage
  \fi
  \@mainmatterfalse
}%
\ifx\undefined\cleartorecto
 \def\cleartorecto{\cleardoublepage}%
\fi
```

Unnumbered tables

`unnumtable` (*env.*) An unnumbered table does not float.

```
\def\@to{to}%
\newenvironment{unnumtable}{%
 \par
 \addpenalty\predisplaypenalty
 \addvspace\abovedisplayskip
 \hbox\@to\hsize\bgroup\hfil\ignorespaces
  \let\@Hline\@empty
}{%
 \unskip\hfil\egroup
 \penalty\postdisplaypenalty
 \vskip\belowdisplayskip
 \aftergroup\ignorespaces
 \@endpetrue
}%
```

Emulate `\toprule` and friends

```
\providecommand\toprule{\hline\hline}%
\providecommand\colrule{\\\hline}%
\providecommand\botrule{\\\hline\hline}%
```

Define sectioning command below `\subsubsection`.

```
\DeclareRobustCommand\subsubsubsection{%
 \@startsection{subsubsection}{4}%
  {\z@}{-15\p@\@plus-5\p@\@minus-2\p@}%
  {5\p@}{\normalfont\normalsize\itshape}%
}%
```

5.6 In lieu of `ltxdoc.cfg`

We don't want everything to appear in the index

```
\DoNotIndex{\',\.,\@M,\@@input,\@Alph,\@alph,\@addtoreset,\@arabic}
\DoNotIndex{\@badmath,\@centercr,\@cite}
\DoNotIndex{\@dotsep,\@empty,\@float,\@gobble,\@gobbletwo,\@ignoretrue}
\DoNotIndex{\@input,\@ixpt,\@m,\@minus,\@mkboth}
\DoNotIndex{\@ne,\@nil,\@nomath,\@plus,\roman,\@set@topoint}
\DoNotIndex{\@tempboxa,\@tempcnta,\@tempdima,\@tempdimb}
\DoNotIndex{\@tempswafalse,\@tempswatrue,\@viipt,\@viiipt,\@vipt}
\DoNotIndex{\@vpt,\@warning,\@xiipt,\@xipt,\@xivpt,\@xpt,\@xviipt}
\DoNotIndex{\@xxpt,\@xxvpt,\\,\ ,\addpenalty,\addtolength,\addvspace}
\DoNotIndex{\advance,\ast,\begin,\begingroup,\bfseries,\bgroup,\box}
\DoNotIndex{\bullet}
\DoNotIndex{\cdot,\cite,\CodelineIndex,\cr,\day,\DeclareOption}
\DoNotIndex{\def,\DisableCrossrefs,\divide,\DocInput,\documentclass}
\DoNotIndex{\DoNotIndex,\egroup,\ifdim,\else,\fi,\em,\endtrivlist}
\DoNotIndex{\EnableCrossrefs,\end,\end@dblfloat,\end@float,\endgroup}
\DoNotIndex{\endlist,\everycr,\everypar,\ExecuteOptions,\expandafter}
\DoNotIndex{\fbox}
\DoNotIndex{\filedate,\filename,\fileversion,\fontsize,\framebox,\gdef}
\DoNotIndex{\global,\halign,\hangindent,\hbox,\hfil,\hfill,\hrule}
\DoNotIndex{\hsize,\hskip,\hspace,\hss,\if@tempswa,\ifcase,\or,\fi,\fi}
\DoNotIndex{\ifhmode,\ifvmode,\ifnum,\iftrue,\ifx,\fi,\fi,\fi,\fi,\fi}
\DoNotIndex{\input}
\DoNotIndex{\jobname,\kern,\leavevmode,\let,\leftmark}
\DoNotIndex{\list,\llap,\long,\m@ne,\m@th,\mark,\markboth,\markright}
\DoNotIndex{\month,\newcommand,\newcounter,\newenvironment}
\DoNotIndex{\NeedsTeXFormat,\newdimen}
\DoNotIndex{\newlength,\newpage,\nobreak,\noindent,\null,\number}
\DoNotIndex{\numberline,\OldMakeindex,\OnlyDescription,\p@}
\DoNotIndex{\pagestyle,\par,\paragraph,\paragraphmark,\parfillskip}
\DoNotIndex{\penalty,\PrintChanges,\PrintIndex,\ProcessOptions}
\DoNotIndex{\protect,\ProvidesClass,\raggedbottom,\raggedright}
\DoNotIndex{\refstepcounter,\relax,\renewcommand}
\DoNotIndex{\rightmargin,\rightmark,\rightskip,\rlap,\rmfamily}
\DoNotIndex{\secdef,\selectfont,\setbox,\setcounter,\setlength}
\DoNotIndex{\settowidth,\sfcode,\skip,\sloppy,\slshape,\space}
\DoNotIndex{\symbol,\the,\trivlist,\typeout,\tw@,\undefined,\uppercase}
\DoNotIndex{\usecounter,\usefont,\usepackage,\vfil,\vfill,\viiipt}
\DoNotIndex{\viipt,\vipt,\vskip,\vspace}
\DoNotIndex{\wd,\xiipt,\year,\z@}
\DoNotIndex{\next}
```

Direct `ltxdoc` to produce an index.

```
\AtEndDocument{\PrintIndex\PrintChanges}%
```

5.7 Extension to LaTeX's `filecontents` Environment

We want to coax the version number into `filecontents`-generated files. Note that we expect `\fileversion` and `\filedate` to hold the needed information. For this to be the case, your document should execute the `\GetFileInfo` command (as documented in section 4.1.3) before any instances of `filecontents`.

```
\makeatletter
\def\endfilecontents{%
 \immediate\write\reserved@c{%
  \string\iffalse\space ltxdoc klootch^^J%
  \ifx\undefined\fileversion\else
  \ifx\undefined\filedate\else
  This file has version number \fileversion, last revised \filedate.%
  \fi\fi
  \string\fi
 }%
 \immediate\closeout\reserved@c
 \def\T##1##2##3{%
  \ifx##1\@undefined\else
    \@latex@warning@no@line{##2 has been converted to Blank ##3e}%
  \fi
 }%
 \T\L{Form Feed}{Lin}%
 \T\I{Tab}{Spac}%
 \immediate\write\@unused{}%
}%
\expandafter\let\csname endfilecontents*\endcsname\endfilecontents
\makeatother
```

Alter formatting in arrays; set them tight.

```
\setlength\arraycolsep{0pt}%
```

5.8 End of the `kernel` DOCSTRIP module

```
%</kernel>
```

5.9 Tail of the `package` DOCSTRIP module

Here is the remainder of the package code.

```
%<*package>
```

Currently, there is little.

```
%</package>
```

6 Font Package for Acrobat Compatability

The package `acrofont` substitutes Acrobat-standard fonts for Computer Modern where possible, even in math mode. Documents typeset with this package in effect will require as little downloaded font data as possible, but will not be exemplars of fine math typesetting.

6.1 Beginning of the `fonts` DOCSTRIP module

The document class module comprises this and the next four sections.

`\class@base` We define in exactly one spot the base class. Typically that class will be one of `book`, `article`, or `report`. The base class effectively defines the use and the markup scheme of the class of documents to be handled by this class.

This class is a variant of the standard LaTeX book class: `ftp://ctan.tug.org/tex-archive/macros/latex/unpacked`.

```
%<*fonts>
\def\class@name{ltxdocext}%
\expandafter\ClassInfo\expandafter{\class@name}{%
 Written for \protect\LaTeXe\space
 by A. Ogawa (arthur_ogawa at sbcglobal.net)%
}%
```

6.2 Variants on psfonts packages

The following uses `times.sty` from `/packages/psnfss/psfonts.dtx`

```
\RequirePackage{times}%
```

The following uses `mathptm.sty` from `/packages/psnfss/psfonts.dtx`

```
\RequirePackage{mathptm}%
```

The following is a customization of `ot1ptmcm.fd`. The virtual font referred to here `zptmnocmr` is a variant of Sebastian Rahtz's `zptmcmr`, but with even more glyphs moved from CM to Acrobat-standard fonts.

```
\DeclareFontFamily{OT1}{ptmcm}{}
\DeclareFontShape{OT1}{ptmcm}{m}{n}{
   <-> zptmnocmr
}{}
\DeclareFontShape{OT1}{ptmcm}{l}{n}{<->ssub * ptmnocm/m/n}{}
```

The following is a customization of `omlptmcm.fd` The virtual font referred to here `zptmnocmrm` is a variant of Sebastian Rahtz's `zptmcmrm`, but with even more glyphs moved from CM to Acrobat-standard fonts.

```
\DeclareFontFamily{OML}{ptmcm}{\skewchar \font =127}
\DeclareFontShape{OML}{ptmcm}{m}{it}{
   <-> zptmnocmrm
}{}
\DeclareFontShape{OML}{ptmcm}{l}{it}{<->ssub * ptmcm/m/it}{}
\DeclareFontShape{OML}{ptmcm}{m}{sl}{<->ssub * ptmcm/m/it}{}
\DeclareFontShape{OML}{ptmcm}{l}{sl}{<->ssub * ptmcm/m/sl}{}
```

The following is a customization of `omspzccm.fd` The virtual font referred to here `zpzcnocmry` is a variant of Sebastian Rahtz's `zpzccmry`, but with even more glyphs moved from CM to Acrobat-standard fonts.

```
\DeclareFontFamily{OMS}{pzccm}{}
\DeclareFontShape{OMS}{pzccm}{m}{n}{
   <-> zpzcnocmry
}{}% cmsy10 Symbol Zapf Chancery Medium-Italic Times-Roman
\DeclareFontShape{OMS}{pzccm}{l}{n}{<->ssub * pzccm/m/n}{}
```

The following is a customization of `omxpsycm.fd` The virtual font referred to here `zpsynocmrv` is a variant of Sebastian Rahtz's `zpsycmrv`, but with even more glyphs moved from CM to Acrobat-standard fonts.

```
\DeclareFontFamily{OMX}{psycm}{}
\DeclareFontShape{OMX}{psycm}{m}{n}{
   <-> zpsynocmrv
}{}
\DeclareFontShape{OMX}{psycm}{l}{n}{<->ssub * psycm/m/n}{}
%
\DeclareFontEncoding{8r}{}{}% from file: 8renc.def
\DeclareFontFamily{8r}{cmr}{\hyphenchar\font45 }% from file: 8rcmr.fd
\DeclareFontShape{8r}{cmr}{m}{n}{
 <-> cmr10
}{}
```

6.3 Font definition files

The following forces LaTeX to do now what it would do anyway: load the 'font definition' information for the fonts that we use. In this way, we prepare for faster processing through the `\dump` of a preformatted macro package that will not need to read in any packages or font definitions from disk.

```
\input{8rphv.fd}%
\input{8rptm.fd}%
\input{ot1phv.fd}%
\input{ot1ptm.fd}%
\input{t1ptm.fd}%
```

6.4 More math substitutions

The following definitions arrange to get certain glyphs from the text font instead of out of math pi fonts. In particular, the copyright and registered symbols are single glyphs instead of composites involving the big circle from the `cmsy` font.

```
 \def\eightRChar#1{{\def\encodingdefault{8r}\fontencoding\encodingdefault\selectfont\char"#1}}
 \def\LANGLE{$<$}%{\eightRChar{8B}}%
 \def\RANGLE{$>$}%{\eightRChar{9B}}%
%\def\ASTER{\eightRChar{2A}}%
%\def\DAGGER{\eightRChar{86}}%
%\def\DDAGGER{\eightRChar{87}}%
 \def\BULLET{\eightRChar{95}}%
%\def\SECTION{\eightRChar{A7}}%
%\def\PARAGRAPH{\eightRChar{B6}}%
 \def\VERTBAR{\eightRChar{7C}}%
 \def\COPYRIGHT{\eightRChar{A9}}%
 \def\REGISTERED{\eightRChar{AE}}%
 \def\textbar{\VERTBAR}%
 \def\textbullet{\BULLET}%
 \def\textcopyright{\COPYRIGHT}%
 \def\textregistered{\REGISTERED}%
```

I have removed `\ensuremath` from the following definition, and all commands like `\mathsection`have been converted to e.g., `\textsection`.

```
\def\@makefnmark{\@thefnmark}%
```

```
\def\@fnsymbol#1{{\ifcase#1\or *\or \dagger\or \ddagger\or
   \textsection\or \textparagraph\or \|\or **\or \dagger\dagger
   \or \ddagger\ddagger \else\@ctrerr\fi}}
```

6.5 End of the `fonts` DOCSTRIP module

Here ends the module.

```
%</fonts>
```

7 Programming Conventions

In writing the above code, I cleave to certain conventions, noted here. My goal in explaining them is to encourage others maintaining this body of code to consider following them as well. The benefits are twofold: Some of the coding conventions aim to avoid programming pitfalls; following them reduces maintenance costs. Others make the code easier to read; following these eases the process of understanding how the code works.

And, for my part, I prefer to read code of this type.

7.1 Whitespace Conventions

Exactly where code lines break and indent, and where additional whitespace is inserted is explained here.

- Each new macro definition or register assignment begins a new line. Therefore, `\def`, `\newcommand`, and their ilk will start in column 1.
- Code is indented one space for each level of nesting within braces `{}`.
- Likewise, if possible, for `\if`... and matching `\fi`.
- However, the closing brace or `\fi` is outdented by one so that it falls at the same level of indentation as its matching brace or `\if`, and it appears alone on its line.
- Use of the `tab` character is deprecated (tabs are not standardized across all applications and operating systems).
- Lines of code are limited to 72 characters. I follow this convention mostly to ease the transmission of files via email (a deprecated practice) and to accomodate people with small monitors. But `ltxdoc` output looks better with the shorter lines, too.
- Extraneous whitespace in the replacement part of a macro definition is avoided by using the comment character `%`. In most cases, if falling at the end of a line of code, a brace will be immediately followed by a comment character, as will the macro parameter `#1` and any one-letter control sequence, like `\\`.
- Extraneous whitespace in the package file is also avoided. When TEX reads in the .sty file, it will process `\def`s, and other commands, but will not process blank spaces. This practice is simply a discipline. You don't need

to do this. But sometimes TEX has to read in a file while it is in horizontal mode, at which point this practice is essential.

These conventions taken together are illustrated by the following:

```
%\def\prepdef#1#2{%
% \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
% \toks@ii{#2}%
% \edef#1{\the\toks@ii\the\toks@}%
%}%
%
```

In the above, the definition of `\prepdef` would not fit on a single line, and required breaking. The first and last lines have matching braces, and are a the same level of indentation, with the last line containing a single brace.

Each of the three intervening lines has balanced braces and is indented by one space. Each line that would otherwise end in a single brace character is terminated by a comment character.

Some coders rely on the fact that a space character seen by TEX's scanner while in vertical mode is a no-op. Be that as it may, I eliminate them unless actually intentional.

7.2 Conventions For Procedures

Here are some of my preferences when writing procedures:

- I dislike defining a macro within another macro, especially when the pattern part is non-nil. You know, you are not saving much space in `mem` when you do this, right? You do save space in the hash table and the string pool, though. On the other hand, we are not dealing with small TEX engines anymore; Team LATEX has made sure of this.
- If two or more macros have very similar replacement parts, consider layering.
- Macros may perform parsing, may maintain tokens or registers, or may set type (produce marks). I try to avoid combining the three functions in a single macro.
- When a procedure both does assignments and sets type, I try to have a clean separations between the two activities. Try to avoid:

  ```
  % \vskip10pt
  % \parindent=0pt
  % \leavevmode
  %
  ```

- The Boolean calculus (cf. `\@ifx`) is very useful and produces code that executes nicely. Using it also helps avoid your having to debug problems where `\if...` and `\fi` are not properly balanced (a nightmare, in case you have not already experienced it).

7.3 Conventions For LaTeX

Team LaTeX make certain recommendations in `clsguide.tex`. Ones that I particularly pay attention to are:

- For the sake of "color safety", use `\sbox` rather than `\setbox`, `\mbox` rather than `\hbox`, and `\parbox` or `minipage` rather than `\vbox`.
- Use `\newcommand` and `\newenvironment` to declare user-level commands and environments. Avoid the idiom `\def\foo`, `\def\endfoo` to define an environment. Ideally, all user-level markup could be extracted from the document class by grepping on `\newcommand` and `\newenvironment`.
- Prefer to use `\setlength` to assign registers.

I cannot help but notice that much of the code of LaTeX itself fails to comply with many of the coding recommendations of Team LaTeX.

Change History

Index

D

E

M

N

O

P

Q

R

S

T

U

V

W

Z